Critiques for Progressive Meshes

An Nguyen
CS 448B, Winter 2000

February 7, 2000

1 Citation for the paper

Hugues Hoppe, *Progressive Meshes*, Proc. Siggraph 1996, pp. 99-108.

2 Abstract of the paper

The paper introduces the progressive mesh representation as a scheme for storing and transmitting arbitrary triangle meshes. It provides a solution to several practical problems in graphics: smooth geomorphing, progressive transmission, mesh compression, and selective refinement. The paper also presents a mesh simplification procedure for constructing an progressive presentation from an arbitrary mesh.

3 Problem definition

Input: A mesh, as a tuple $M = (K, V, D, S)$ where V specifies the geometry, K the set of faces, D the set of discrete attributes associated with the faces, and S the set of scalar associated with the corners of K.
Output: A representation of the mesh

4 Summary

The idea behind the progressive mesh is the *edge collapse* operation. This operation collapses an edge on the mesh into a single vertex. Each edge collapse operation is invertible. Its inverse, the *vertex split* operation, splits a vertex into two vertices.

A mesh can be simplified into a coarser mesh by applying a sequence of edge collapse transformation. Conversely, a mesh can be obtained by applying vertex splits on its coarser mesh. The information needed for these vertex splits together with the coarse map form a progressive mesh representation for the original mesh.

One of the main issue of constructing a progressive mesh is the how to choose the sequence of edges to collapse. An obvious clear choice for progressive mesh would be picking the edges to be collapsed at random. While this may give an progressive mesh, the quality of the intermediate mesh may be poor. Another issue is the computation of the attributes at the new vertex arisen from the edge collapse.

The author proposes a method in which an edge minimizing some mesh energy is collapsed at each step. The mesh energy is of the form:

$$E(M) = E_{dist}(M) + E_{spring}(M) + E_{scalar}(M) + E_{disc}(M)$$

where

- E_{dist} and E_{spring} measure how well the mesh after the edge collapse approximates the original mesh. E_{spring} is used as a regularized term for the optimization, it is most important at the beginning of the construction.
- E_{scalar} measures how well the scalar attributes of the resulting mesh approximate the original attributes.
- E_{disc} preserves the geometry of the discontinuity curves. If an edge collapse changes the topology of discontinuity curves, the edge collapse is either disallowed or penalize.

The minimization procedure first solves the minimization problem for $E_{dist} + E_{spring}$ to find the location of the new vertex created from the collapsed edge. This process is similar to previous work by Hoppe et. al. in mesh optimization. The scalar attributes of the new vertex can then be obtained by solving a minimization problem for E_{scalar}. The penalty associated with E_{disc} is added on top of the optimal value of $E_{dist} + E_{spring}$ which only affects the edges to be collapsed next.

5 Significance of the results and comments

This is the first time progressive mesh is introduced. Progressive representation is very versatile in graphics.

- Progressive mesh can be used for mesh simplification. It contains a spectrum of approximations of a mesh at many different levels of details. Users thus have the option to trade off between accuracy and size of the mesh approximation.
- Meshes in the progressive mesh can be transformed smoothly from one into another. Progressive mesh thus allows smooth switching of the mesh between different levels of details.
- Progressive mesh preserves not just the geometry but also the discrete and scalar attributes associated with the mesh.

- Progressive mesh can be used to transmit a large mesh progressively
- Progressive mesh allows selective refinement where details is added to the model only in desired areas, for example, near the view point.
- Progressive mesh can be used as a mesh compression since it exploits well the locality of mesh attributes, and property of planar graph. This is surprising, since progressive mesh encodes a family of meshes.

Progressive mesh clearly is a powerful tool. It has some small short coming, though. Since the edge collapse operation preserves the topology of the mesh surface, if the original surface is complicated topologically, progressive mesh may not be abbe to reduce the size of the mesh all the way down to some small number. Progressive algorithm does allow for topology change, yet the procedure for change here is probably too ad hock to deal with discontinuity in general.

This paper is an excellent practical paper. The author implemented and tested out the idea for some meshes. The tests illustrate well the power of progressive mesh. As an example, in one of the test case, an airplane mesh with 13,546 faces are well approximated by that with 1000 faces. He also argues and illustrates via an example that progressive mesh yields better approximation than Multiresolution Analysis, another mesh simplification procedure.

[I find it's interesting that two mesh examples given involved mesh derived from images. The result of the algorithm is pretty good.]

There are minor issues that worth noticed. First, the running time of the algorithm is high. The program currently runs offline because it's too slow. Secondly, the algorithm seems to contain a lot of parameters to be fine tuned. The author doesn't describe all the details.

6 Discussion

1. Although progressive method is a solution to many practical problem, is progressive mesh practical to build? In particular, would it be practical to spend 16-106 minutes to build the progressive mesh? If so, when?
2. The energy measure discussed in the paper has several terms, each of different units. Is it an issue?